

Machine Learning Engineer Nanodegree

Capstone Proposal

Abd Elrahman adel elsayied

11 April, 2019

Proposal

Domain Background:

AI (Artificial Intelligence) and machine learning is an important fields that helps in solving many problems like computer vision and self-driving cars. Recognizing objects in images and classifying it is an important task because it helps organizations to fasten their work to save its time.

In this project I have deep neural networks to train 40 distinct face. This can help in monitoring (attendance, security) systems and it could be part of bigger systems.

The project is important because it can save time to check each person identity and it's more accurate

It can apply in any work place or school or college.

Problem Statement:

In paper systems it take a lot of time to record the attendance of students or employ. So I want to make a deep learning model that can identify every person (student, employ) in the place.

I have got the [dataset](#) from kaggle has 400 labeled images. I want to make a classifier that can take an image and predict person identity.

Datasets and Inputs:

I will use a dataset I have got from kaggle which contains 400 face images 40 distinct person.

The content of dataset:

- Total number of the 40's person images: 400.
- Train set: 280.
- Valid set: 80.
- Test set: 40.
- Number of classes: 40 ? Image size: The original dataset consisted of 92 x 112, while I reshape it to be 299 x299

A preview for the 40 distinct faces in database:



I have used a train files shape as (280, 299, 290, 3) and a face label shape (280, 40) to input it to Keras CNN.

I will split the training set into training and validation set to maintain the balance of the dataset.

The data contains 6 images for each person as a training set (total=240) and 4 as a test for each type (total=160).

Solution Statement:

The solution for this problem is to train a neural network model that can classify new inputted images to one of the 40 face.

I will use a Convolutional neural networks CNN to classify these images and to get a good accuracy.

Benchmark Model:

I will benchmark my model with a model in [kaggle's](#) kernel .

Model's structure:

```
model = Sequential()

model.add(Conv2D(filters = 20, kernel_size = (5,5),padding = 'Same',
                  activation = 'relu', input_shape = (64,64,1)))

model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.25))
```

```
model.add(Conv2D(filters = 50, kernel_size = (6,6),padding = 'Same',  
                activation = 'relu'))  
  
model.add(MaxPool2D(pool_size=(2,2)))  
model.add(Dropout(0.25))  
  
model.add(Conv2D(filters = 150, kernel_size = (5,5),padding = 'Same',  
                activation = 'relu', input_shape = (64,64,1)))  
  
model.add(Flatten())  
model.add(Dense(256, activation = "relu"))  
model.add(Dropout(0.5))  
model.add(Dense(40, activation = "softmax"))
```

with accuracy 97.5%, I'm looking to approach to this accuracy.

Evaluation Metrics:

I will use accuracy test score on the test set to evaluate my model because it is appropriate if the dataset is balanced.

Project Design:

1. Import Dataset.
2. Preprocess Dataset. I will try to resize and normalize the images.
3. Making model architecture. I will try using CNN model from scratch and a pertained model like InceptionV3 or Resnet50.
4. Train the model.
5. Test the model.